L Number	Hits	Search Text	DB	Time stamp
1	205	(intercept$3 with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1)) and IPC$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:30
2	171	((intercept$3 with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1)) and IPC$1) and monitor$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:30
3	42	(((intercept$3 with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1)) and IPC$1) and monitor$1) and (block$1 with (sender$1 source$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:30
-	3901	719/$.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:29
-	392	719/$.ccls. and ((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:29
-	11	(719/$.ccls. and ((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1))) and (synchron$ with (re$1direct$3 intercept$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:12
-	4	((719/$.ccls. and ((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1))) and (synchron$ with (re$1direct$3 intercept$3))) and source$1 and destination$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:06
-	22595	((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:11
-	630	(((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1))) and ((source$1 sender$1 client$1) near3 (ID identifier$1 identity)) and ((receiver$1 destination$1 server$1) near3 (ID identifier$1 identity))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:12
-	20	((((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1))) and ((source$1 sender$1 client$1) near3 (ID identifier$1 identity)) and ((receiver$1 destination$1 server$1) near3 (ID identifier$1 identity))) and (synchron$ with (re$1direct$3 intercept$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:14
-	60	((((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1))) and ((source$1 sender$1 client$1) near3 (ID identifier$1 identity)) and ((receiver$1 destination$1 server$1) near3 (ID identifier$1 identity))) and (transparent$3 with (re$1direct$3 intercept$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:22

C:\APPS\EAST\Workspaces\09661341.wsp

-	21	(((((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1))) and ((source$1 sender$1 client$1) near3 (ID identifier$1 identity)) and ((receiver$1 destination$1 server$1) near3 (ID identifier$1 identity))) and (transparent$3 with (re$1direct$3 intercept$3))) and (IPC (inter$1process adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:21
-	2	5949876.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:21
-	812	719/$.ccls. and source$1 and destination$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:23
-	58	((transparent$3 with (re$1direct$3 intercept$3)) same (IPC (inter$1process communication))) and source$1 and destination$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:26
-	48	(((transparent$3 with (re$1direct$3 intercept$3)) same (IPC (inter$1process communication))) and source$1 and destination$1) and (ID identity identifier)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:23
-	9	((transparent$3 with (re$1direct$3 intercept$3)) same (IPC (inter$1process communication))) and (synchron$ with (re$1direct$3 intercept$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:27
-	71	((transparent$3 with (re$1direct$3 intercept$3)) same (IPC (inter$1process communication))) and synchron$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:28
-	70	(((transparent$3 with (re$1direct$3 intercept$3)) same (IPC (inter$1process communication))) and synchron$) and (generat$3 creat$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:28
-	19	((((transparent$3 with (re$1direct$3 intercept$3)) same (IPC (inter$1process communication))) and synchron$) and (generat$3 creat$3)) and kernel	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:42
-	437	kernel same (receiv$3 near3 ((IPC adj request$1) request$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:43
-	220	kernel with (receiv$3 near3 ((IPC adj request$1) request$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:44
-	133	(kernel with (receiv$3 near3 ((IPC adj request$1) request$1))) and ((creat$3 generat$3 build$3) with request$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:44
-	37	((kernel with (receiv$3 near3 ((IPC adj request$1) request$1))) and ((creat$3 generat$3 build$3) with request$1)) and ((intercept$3 re$1direct$3) near3 request$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:50

C:\APPS\EAST\Workspaces\09661341.wsp

-	387	((intercept$3 re$1direct$3) with (call$1 method$1 request$1 message$1)) and (IPC (interprocess adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:52
-	189	(((intercept$3 re$1direct$3) with (call$1 method$1 request$1 message$1)) and (IPC (interprocess adj communication))) and ((monitor$3 debug$4) with (application$1 program$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:52
-	4	((((intercept$3 re$1direct$3) with (call$1 method$1 request$1 message$1)) and (IPC (interprocess adj communication))) and ((monitor$3 debug$4) with (application$1 program$1))) and (kernel with extension$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 14:53
-	94	((((intercept$3 re$1direct$3) with (call$1 method$1 request$1 message$1)) and (IPC (interprocess adj communication))) and ((monitor$3 debug$4) with (application$1 program$1))) and kernel	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 15:43
-	4390	(kernel OS (operating adj system)) with (receiv$3 near3 (request$1 (system adj2 call$1) function$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 15:55
-	754	((kernel OS (operating adj system)) with (receiv$3 near3 (request$1 (system adj2 call$1) function$1))) and ((kernel OS (operating adj system)) with ((creat$3 generat$3 build$3) near3 (IPC request$1 function$1 call$1 method$1)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 16:31
-	94	(((kernel OS (operating adj system)) with (receiv$3 near3 (request$1 (system adj2 call$1) function$1))) and ((kernel OS (operating adj system)) with ((creat$3 generat$3 build$3) near3 (IPC request$1 function$1 call$1 method$1)))) and (IPC (inter$1process adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 15:50
-	45	((((kernel OS (operating adj system)) with (receiv$3 near3 (request$1 (system adj2 call$1) function$1))) and ((kernel OS (operating adj system)) with ((creat$3 generat$3 build$3) near3 (IPC request$1 function$1 call$1 method$1)))) and (IPC (inter$1process adj communication))) and (intercept$3 re$1direct$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 15:50
-	7	(((((kernel OS (operating adj system)) with (receiv$3 near3 (request$1 (system adj2 call$1) function$1))) and ((kernel OS (operating adj system)) with ((creat$3 generat$3 build$3) near3 (IPC request$1 function$1 call$1 method$1)))) and (IPC (inter$1process adj communication))) and (intercept$3 re$1direct$3)) and ((debug$ monitor$3) near3 (application$1 program$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 15:51
-	80501	(kernel OS (operating adj system)) with (request$1 (system adj2 call$1) function$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 15:55
-	849	((kernel OS (operating adj system)) with (request$1 (system adj2 call$1) function$1)) and ((intercept$3 re$1direct$3) near3 (kernel OS (operating adj system))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 15:56

-	32	(((kernel OS (operating adj system)) with (request$1 (system adj2 call$1) function$1)) and ((intercept$3 re$1direct$3) near3 (kernel OS (operating adj system)))) and (dynamic$ near4 (intercept$3 re$1direct$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:03
-	7667	((kernel OS (operating adj system)) with ((creat$3 generat$3 build$3) near3 (IPC request$1 function$1 call$1 method$1)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:02
-	15	(((kernel OS (operating adj system)) with ((creat$3 generat$3 build$3) near3 (IPC request$1 function$1 call$1 method$1)))) and (dynamic$ near4 (intercept$3 re$1direct$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 16:35
-	6966	((kernel OS (operating adj system)) with ((modif$3 chang$3 set$4) near3 (IPC request$1 function$1 call$1 method$1)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:03
-	29	(((kernel OS (operating adj system)) with ((modif$3 chang$3 set$4) near3 (IPC request$1 function$1 call$1 method$1)))) and (dynamic$ near4 (intercept$3 re$1direct$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:04
-	744	(((kernel OS (operating adj system)) with ((modif$3 chang$3 set$4) near3 (IPC request$1 function$1 call$1 method$1)))) and (intercept$3 re$1direct$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:31
-	159	((((kernel OS (operating adj system)) with ((modif$3 chang$3 set$4) near3 (IPC request$1 function$1 call$1 method$1)))) and (intercept$3 re$1direct$3)) and ((monitor$3 debug$4) near4 (application$1 program$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:17
-	7	(((((kernel OS (operating adj system)) with ((modif$3 chang$3 set$4) near3 (IPC request$1 function$1 call$1 method$1)))) and (intercept$3 re$1direct$3)) and ((monitor$3 debug$4) near4 (application$1 program$1))) and 719/$.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:15
-	0	(modif$ near3 (IPC adj2 request$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:16
-	8636	(modif$ near3 (call$1 request$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:16
-	617	(modif$ near3 (call$1 request$1)) same (kernel OS (operating adj system))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:26
-	329	((modif$ near3 (call$1 request$1)) same (kernel OS (operating adj system))) and (monitor$3 debug$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:17
-	12	(((modif$ near3 (call$1 request$1)) same (kernel OS (operating adj system))) and (monitor$3 debug$4)) and IPC	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:17

-	31	(((modif$ near3 (call$1 request$1)) same (kernel OS (operating adj system))) and (monitor$3 debug$4)) and (IPC (inter$1process adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:31
-	1914	((modif$ chang$3 set$4) near3 (call$1 request$1)) with (kernel OS (operating adj system))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:26
-	113	(((modif$ chang$3 set$4) near3 (call$1 request$1)) with (kernel OS (operating adj system))) and (IPC (inter$1process adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:31
-	61	((((modif$ chang$3 set$4) near3 (call$1 request$1)) with (kernel OS (operating adj system))) and (IPC (inter$1process adj communication))) and (intercept$3 re$1direct$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/18 17:32
-	3	(synchron$ near3 (IPC (inter$1process adj communication))) and ((intercept$3 re$1direct$3) near4 request$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:25
-	5	(synchron$ adj2 (IPC (inter$1process adj communication))) and ((intercept$3 re$1direct$3) with (IPC request$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:54
-	259	(IPC (inter$1process adj communication)) and ((intercept$3 re$1direct$3) with (IPC request$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:28
-	1	((IPC (inter$1process adj communication)) and ((intercept$3 re$1direct$3) with (IPC request$1))) and ("L4" near3 (OS (operating adj system)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:32
-	1	"6389540".PN.	USPAT	2004/02/19 09:29
-	1	"6330677".PN.	USPAT	2004/02/19 09:30
-	1	"6308317".PN.	USPAT	2004/02/19 09:31
-	1	"6295607".PN.	USPAT	2004/02/19 09:31
-	1	"6282652".PN.	USPAT	2004/02/19 09:31
-	1	"6182226".PN.	USPAT	2004/02/19 09:32
-	1	"6003134".PN.	USPAT	2004/02/19 09:32
-	1	"6026237".PN.	USPAT	2004/02/19 09:32
-	1	((IPC (inter$1process adj communication)) and ((intercept$3 re$1direct$3) with (IPC request$1))) and (((IPC (inter$1process adj communication)) and ((intercept$3 re$1direct$3) with (IPC request$1))) and ("L4" near3 (OS (operating adj system)))) and (OS (operating adj system)) and kernel	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:39
-	28	((IPC (inter$1process adj communication)) and ((intercept$3 re$1direct$3) with (IPC request$1))) and (iPC near3 option$1) and (OS (operating adj system)) and kernel	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:41

-	28	((IPC (inter$1process adj communication)) and ((intercept$3 re$1direct$3) with (IPC request$1))) and (iPC near3 option$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:42
-	36	(iPC near3 option$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:42
-	25	(synchron$ with (IPC (inter$1process adj communication))) and ((intercept$3 re$1direct$3) with (IPC request$1 message$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 09:55
-	24	((synchron$ with (IPC (inter$1process adj communication))) and ((intercept$3 re$1direct$3) with (IPC request$1 message$1))) and (kernel OS (operating adj system))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:00
-	7968	(kernel OS (operating adj system)) with (creat$3 build$3 set$4 generat$3 append$3 chang$3) with (request$1 message$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:01
-	493	((kernel OS (operating adj system)) with (creat$3 build$3 set$4 generat$3 append$3 chang$3) with (request$1 message$1)) and ((intercept$3 re$1direct$3) near3 (request$1 message$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:02
-	56	(((kernel OS (operating adj system)) with (creat$3 build$3 set$4 generat$3 append$3 chang$3) with (request$1 message$1)) and ((intercept$3 re$1direct$3) near3 (request$1 message$1))) and (IPC (inter$1process$ adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:05
-	45	kernel same monitor$1 and IPC	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 14:14
-	217859	signal adj processing	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 16:22
-	21	(signal adj processing adj2 system) and framework and (OO (object$1 adj oriented))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 16:23
-	2	6424991.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 16:33
-	2	6195791.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 16:33
-	2	6308314.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 16:34
-	331	((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication))) and source and destination and kernel	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 15:03

-	277	(((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication))) and source and destination and kernel) and monitor$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 14:50
-	1	((((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication))) and source and destination and kernel) and monitor$3) and ((chang$3 modif$3) with (source near3 (ID identity identifier)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 14:51
-	1	((((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication))) and source and destination and kernel) and monitor$3) and ((chang$3 modif$3) with (source with (ID identity identifier)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 14:51
-	136	((((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication))) and source and destination and kernel) and monitor$3) and ((chang$3 modif$3) with source)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 14:52
-	54	(((((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication))) and source and destination and kernel) and monitor$3) and ((chang$3 modif$3) with source)) and ((chang$3 modif$3) with destination)) and (IPC (inter$1process adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 14:58
-	19	((((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication))) and source and destination and kernel) and monitor$3) and 719/$.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 15:02
-	420	719/$.ccls. and (IPC (inter$1process adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 15:03
-	25	(719/$.ccls. and (IPC (inter$1process adj communication))) and ((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 15:04
-	16	((719/$.ccls. and (IPC (inter$1process adj communication))) and ((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication)))) and kernel	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 15:04
-	79	(((((intercept$3 re$1direct$1) with (message$1 request$1 IPC (inter adj process adj2 communication))) and source and destination and kernel) and monitor$3) and ((chang$3 modif$3) with source)) and ((chang$3 modif$3) with destination)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 15:28
-	146	(transparent$3 with (re$1direct$3 intercept$3)) same (IPC (inter$1process communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 16:04
-	633	719/$.ccls. and (intercept$3 re$1direct$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 16:05

-	0	(719/$.ccls. and (intercept$3 re$1direct$3)) and (synchronous$3 with (IPC (inter$1process adj communication)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 16:05
-	4	(719/$.ccls. and (intercept$3 re$1direct$3)) and (synchronous$3 same (IPC (inter$1process adj communication)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 16:06
-	20	(719/$.ccls. and (intercept$3 re$1direct$3)) and synchronous$3 and (IPC (inter$1process adj communication))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/17 16:06
-	2	6330677.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/18 16:09
-	33	719/$.ccls. and ((intercept$3 re$1direct$3) with (IPC (inter$1process adj2 call$1) function$1 method$1 call$1 request$1)) same monitor$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:01
-	1	"6252589".PN.	USPAT	2004/09/10 11:00
-	1	"6247054".PN.	USPAT	2004/09/10 11:00
-	1	"4831518".PN.	USPAT	2004/09/10 11:01
-	2	IPC$1 and monitor$1 and block$3 and unblock$3 and (synchronous near3 semantic$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:03
-	0	IPC$1 and monitor$1 and (block$3 near3 (sender$1 source$1)) and (unblock$3 near3 (sender$1 source$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:04
-	1	IPC$1 and monitor$1 and (block$3 with (sender$1 source$1)) and (unblock$3 with (sender$1 source$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/09/10 11:04

CiteSeer Find: IPC L4 monitor synchronous block Documents Citations

Searching for **PHRASE ipc monitor synchronous block source sender**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB DBLP
No documents match Boolean query. Trying non-Boolean relevance query.
500 documents found. **Only retrieving 125 documents (System busy - maximum reduced). Order: relevance to query.**

RT-IPC: An IPC Extension for Real-Time Mach - Takuro Kitayama (1993) (Correct) (10 citations)
RT-**IPC**: An **IPC** Extension for Real-Time Mach Takuro
www.cs.cmu.edu/afs/cs/project/rtmach/public/papers/ipc93.ps

Debugging a Parallel Program: Capturing Inter-Processor.. - Thomas Gross (1992) (Correct) (1 citation)
Building a special-purpose hardware performance **monitor** is too costly in most scenarios, and the use of
to receive C's message first, processor A will **block**, waiting to receive the message from C, and
www.cs.cmu.edu/afs/cs.cmu.edu/project/iwarp/archive/fx-papers/dw92.ps

High-Level Views of Distributed Executions - Kunz (1995) (Correct) (4 citations)
by Lamport[21]and originally defined for **IPC** events in asynchronous systems only. This relation
helpful during the construction, debugging, and **monitor**ing of distributed applications as well as in
dynamically and communicate and synchronize by **synchronous** and asynchronous message passing. In other
ccnga.uwaterloo.ca/pub/papers/Ps/conf09.ps.Z

Synchronization Overhead Reduction in Timed Cosimulation - Yoo, Choi (Correct) (2 citations)
optimistic timed cosimulation and the other is non-**IPC** (interprocess communication) timed cosimulation.
Communication protocols are classified into **synchronous** protocols such as polling and asynchronous
poppy.snu.ac.kr/Codesign/../papers/hldvt.ps

Improving IPC by Kernel Design - Liedtke (1993) (Correct) (82 citations)
Improving **IPC** by Kernel Design Jochen Liedtke German National
os.inf.tu-dresden.de/papers_ps/jochen/Ipcsosp.ps

Synchronous, Asynchronous, and Causally Ordered Communication - Charron-Bost, Mattern, Tel (1995) (Correct) (1 citation)
ordered (e.g.to realize a causally consistent **monitor** or causal memory [1]or if causally ordered
Synchronous, asynchronous, and causally ordered
Examples are selective receive statements, which **block** the receiver until a suitable message is
www.isa.informatik.tu-darmstadt.de/VS/Publikationen/papers/syn_asy.ps

An Engineering Environment for Hardware/Software Co-Simulation - Becker, Singh, Tell (1992) (Correct) (2 citations)
programs that use Unix interprocess communication (**IPC**) mechanisms to interact with the hardware
the data flow with control and status registers. **Monitor** software, executing on the PXPL5 host
ftp.cs.unc.edu/pub/projects/codesign/dac_cosim92.ps.Z

Test Report of the Inter Process Communication package for.. - Authors Serguei (Correct)
: Serguei Kolos Keywords :Test, Unit Test, ILU, **IPC** Abstract This document describes the results of
Time for obtaining a list of servers 5. Time for **synchronous** and asynchronous method invocations
atddoc.cern.ch/Atlas/Notes/../postscript/Note123.ps

The Persistent Relevance of IPC Performance: New.. - Hsieh, Kaashoek, Weihl (1993) (Correct) (5 citations)
1 The Persistent Relevance of **IPC** Performance: New Techniques for Reducing the **IPC**
execute as Active Messages Active Messages that **block** on a lock or that execute for too long would
www.pdos.lcs.mit.edu/~kaashoek/papers/ipc.ps

Medium Access Control for Synchronous Traffic in the AMNET LAN - Goodall, Burston (Correct)
Medium Access Control For **Synchronous** Traffic In The Amnet Lan David Goodall
latency involved in getting real-time data from a **source** device to the LAN is minimised, thus simplifying
cell providing the slot is freed in time for the **sender** of the **synchronous** cell to use it. This supports
ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/9501.ps.Z

Iterative Joint Design of Fixed-Rate Source Codes and.. - Goldsmith, Effros (1997) (Correct)
source code dimension and infinite channel code **block** length. Shannon theory does not provide any
Iterative Joint Design of Fixed-Rate **Source** Codes and Multiresolution Channel Codes Andrea
www.cco.caltech.edu/~rjm/effros/papers/ct97.ps.Z

The Case For Reliable Concurrent Multicasting Using.. - Levine, Lavo.. (1996) (Correct) (38 citations)
dissemination of information from multiple **sources** to all the members of a group. Furthermore, it
based on feedback from receivers as to whether the **sender** can erase data from memory. In practice,
all generic protocols can be found in [10]2.1 **Sender**-Initiated Protocols A **sender**-initiated reliable
www.cse.ucsc.edu/research/ccrg/publications/brian.mm96.ps.gz

Reducing State Loss For Effective Trace Sampling of.. - Thomas Conte (1996) (Correct) (17 citations)
is the mean retired instructions per cycle (**IPC**)Consider a processor running a benchmark which
clusters are checked against the full trace to **monitor** the sample's representativeness of the
of the instruction frequencies, basic-**block** densities, and cache statistics. If the
www.ece.ncsu.edu/tinker/conte/iccd96.ps

How to Sign Digital Streams - Gennaro, Rohatgi (1997) (Correct) (49 citations)
for an elementary data stream to be multiplexed **synchronously** with the packetized audio and video streams.
One type of solution splits the stream in **blocks**. The **sender** signs each individual **block** and the
theory.lcs.mit.edu/pub/people/rosario/stream.ps.Z

Image Subband-Coding Using an Information-Theoretic Subband.. - Ulug Bayazit (Correct)
of the **source** itself. It is then clear that optimum **block** or **block** transform coding in the rate-distortion
It has been proved recently that for Gaussian **sources** with memory an ideal subband split will produce
ipl.rpi.edu/publications/pearlman_papers/ist-spie95_bp.ps.gz

Bounding Application-to-Application Delays for Multimedia.. - Fang Feng (Correct)
message delays at the application level. The **synchronous** server is designed to control application
of a message to be violated since it may be **block**ed by the messages ahead of it in the FIFO queue.
provided by Cisco Systems, Inc. We modified the **source** code of the FDDI device driver release 1.2 for
www.cs.tamu.edu/research/realtime/feng-mmcn-96.ps.gz

C++ Wrappers for Efficient, Portable, and Flexible Network.. - Schmidt (Correct)
Ipc Sap CWrappers For Efficient, Portable, And
demultiplexing, stop-and-wait flow control, **synchronous** sendside method invocations, and non-adaptive
UNIX system calls that enable asynchronous I/O, non-**block**ing I/O, and urgent message delivery on Sockets.
siesta.cs.wustl.edu/~schmidt/IPC_SAP-92.ps.gz

A Deep X-Ray Survey Of The Pms Population Of The Upper.. - Sciortino Damiani (Correct)
Catalog, nor in the final analysis of Einstein **IPC** data of the same region, yielding 18 **sources** down
pointed observations analyzed with an innovative **source** detection method based on wavelet transforms (cf.
www.astropa.unipa.it/Library/OAPA_preprints/scocen_nsu.ps.gz

Coding for Computing - Orlitsky, Roche (1998) (Correct)
assume that (1) f(X Y)must be determined for a **block** of many independent (X Y)instances, 2) PX
Cliffs, NJ, 1971. 2] T. Berger. Multiterminal **source** coding. In G. Longo, editor, The Information
Alon Orlitsky y James R. Roche z Abstract A **sender** communicates with a receiver who wishes to
www-ece.ucsd.edu/~alon/papers/cod_com.ps

The Increasing Irrelevance of IPC Performance for.. - Bershad (1992) (Correct) (23 citations)
The Increasing Irrelevance of **IPC** Performance for Microkernel-Based Operating
it took seek one disk track, or copy a 512 byte **block** from a system buffer cache into a user buffer, or
ftp.cs.cmu.edu/project/mach/doc/published/IPCperf.ps

First 20 documents Next 20

Try your query at: Google (CiteSeer) Google (Web) CSB DBLP

Google Web Images Groups News Froogle more »
monitor IPC L4 Search Advanced Search Preferences

Web

Results 1 - 10 of about 639 for monitor IPC L4. (0.40 seconds)

[PDF] Synchronous **IPC** over Transparent **Monitors**
File Format: PDF/Adobe Acrobat - View as HTML
... s **monitor** is notified about the **IPC** delivery. 5 Implementation Issues We consider implementation of the synchronous **IPC** mechanism presented above on the **L4** ...
i30www.ira.uka.de/research/ documents/**l4**ka/synchronous-**ipc**.pdf - Similar pages

[PDF] Flexible Access Control Using **IPC** Redirection
File Format: PDF/Adobe Acrobat - View as HTML
... of clan members to chiefs is static in **L4**. ... However, any **IPC** either to a process outside the clan or ... To **monitor** individual processes, we found it necessary to ...
i30www.ira.uka.de/research/ documents/**l4**ka/**ipc**redirect.pdf - Similar pages
[More results from i30www.ira.uka.de]

Citations: Flexible access control using **IPC** redirection - Jaeger ...
... The most recent iteration of Lava (the **L4** successor) incorporates an **IPC** redirection ... 1999. Synchronous **IPC** over Transparent **Monitors** - Trent Jaeger ...
citeseer.ist.psu.edu/context/1083000/0 - 14k - Cached - Similar pages

Synchronous **IPC** over transparent **monitors**
... of these additional semantics upon system **monitors** as necessary ... efficient implementations of the synchronous **IPC** mechanism upon the **L4** microkernel using ...
portal.acm.org/citation.cfm?id=566726.566765 - Similar pages

The future
... With **L4**, every delicate operation is performed using RPCs. ... The **IPC** redirect mechanism can be very useful, for example to **monitor** applications (for ...
kilobug.free.fr/hurd/pres-en/html/node13.html - 9k - Cached - Similar pages

[PDF] Microsoft PowerPoint - Multiserver2.ppt
File Format: PDF/Adobe Acrobat - View as HTML
... Page 17. Lars Reuther Martin Pohlack Christian Helmuth Marcus Völp TU Dresden Operating Systems Group 17 Microkernel- Based Systems Naming on **L4** Naming on **L4** ...
os.inf.tu-dresden.de/Studium/ KMB/Folien/Multiserver2/Multiserver2.pdf - Similar pages

The **L4** microkernel family - developer's bibliography
... concept that is part of the **L4** version 2 ... Synchronous **IPC** over Transparent **Monitors** (T. Jaeger, JE ... Introduces the Transparent **Monitor** concept, a more flexible ...
os.inf.tu-dresden.de/L4/bib.html - 34k - Cached - Similar pages
[More results from os.inf.tu-dresden.de]

USENIX 2001 Annual Technical Conference Paper
... more coarse-grained so that all objects that would have to be locked before going to sleep are in fact protected by a single **monitor**. ... **IPC**, **L4**/x86, 398, 438. ...
www.usenix.org/events/usenix01/ full_papers/hohmuth/hohmuth_html/ - 70k - Cached - Similar pages

[PDF] Proceedings of the 2001 USENIX Annual Technical Conference
File Format: PDF/Adobe Acrobat
... sleep are in fact protected by a single **monitor**. ... Like **L4**, the Fiasco microkernel al- lows transferring ... virtual-to-physical page mappings via **IPC** between tasks. ...

www.usenix.org/events/usenix01/ full_papers/hohmuth/hohmuth.pdf - Similar pages
[More results from www.usenix.org]

[PDF] Microsoft PowerPoint - lect01.ppt
File Format: PDF/Adobe Acrobat - View as HTML
... **L4** and Microkernels Background Page 18. ... File Socket Pipe Address Space Page ACL Segment Process Task Thread Event **IPC** Semaphore **Monitor** Mutex Priority ...
www.cse.unsw.edu.au/~cs9242/03/lectures/lect01.pdf - Similar pages

New! Get the latest web results on **monitor IPC L4** emailed to you with Google Web Alerts.

Gooooooooooogle ▶
Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

Web

Results **1** - **10** of about **625** for **IPC monitor synchronous semantic**. (**0.54** seconds)

CSGSC 2004
... Microkernels of modern operating systems use **synchronous IPC** semantics for ... the possibility of of misinterpretation of **IPC** timeout events by **monitors**. ...
csgsc.idi.ntnu.no/2004/cgi-bin/ info_presentations_everything.php - 25k - Cached - Similar pages

[PPT] Process Scheduling (Review)
File Format: Microsoft Powerpoint 97 - View as HTML
... seg_3. **monitor**. 9. ... 16. OSARC Fall 2003. **Synchronous** Message Passing. ... 21. OSARC Fall 2003. **IPC** Using Remote Procedure Call (RPC). Suggested by Birell and nelson in ...
dsg.port.ac.uk/teaching/osarc/lectures/lecture04.ppt - Similar pages

[DOC] COLLABORATION COMPONENTS FOR PROGRAMMING REAL-TIME **SYNCHRONOUS** ...
File Format: Microsoft Word 97 - View as HTML
... CVW) has built a tool to enable **synchronous** collaboration among ... is then encoded into the **semantic** template after ... To **monitor** the various network elements it is ...
www.caip.rutgers.edu/~pravinb/SEM/thesis/final.doc - Similar pages

[PDF] Mechanism and Policy of Events and State Machines
File Format: PDF/Adobe Acrobat - View as HTML
... (Also called "**monitor**" in some ... covers the asynchronous and completion parts of the event **IPC**. ... when a new one arrives, while the **synchronous** processing is ...
www.orocos.org/documents/**ipc**.pdf - Similar pages

[PDF] Telecooperation III: Ubiquitous & Mobile Computing
File Format: PDF/Adobe Acrobat - View as HTML
... note: we'll assume that **IPC** directly on ... typed typing 3.5 asynchronous **synchronous** communication-sync. ... user **monitors** coordination order processing manu factu ...
nibbler.tk.informatik.tu-darmstadt.de/ LectureNotes/ss03/TK1/TK1-K3-DistProg-SS03.pdf - Similar pages

Tripos Questions
... and committing 1994/8/2 **IPC** in BSD ... RPC/ORB, event handling (polling, a/**synchronous** callback) 2000 ... of a carpark with and implementation of **monitors** and semaphores ...
www.cl.cam.ac.uk/users/dmr25/superv/questions.html - 17k - Cached - Similar pages

[PDF] page 2 Abstract types, 30 accept() , 65, 66 Acceptor-Connector ...
File Format: PDF/Adobe Acrobat - View as HTML
... IP), 24 Interprocess communication (**IPC**), local and remote ... pattern **Monitor** Object, see **Monitor** Object pattern ... protocols, asynchronous and **synchronous**, 26–28 ...
www.informit.com/content/ images/0201604647/index/schmidtindex.pdf - Similar pages

Citations: The Structuring of Systems Using Upcalls - Clark ...
... fast inter process communication (**IPC**) POSIX specifies ... The application of **synchronous** communication in SDL ... Legacy Transaction Processing **Monitor** - Roger Barga ...
citeseer.ist.psu.edu/context/11873/0 - 28k - Cached - Similar pages

Acronymes Informatique
... CMS. Conversational **Monitor** System (IBM). CODASYL. ... IP. Internet Protocol. **IPC**. Inter-Process Communication. IPX. ... SDLC. **Synchronous** Data Link Control. SDM. ...
www.alyon.org/~byc/scsi/acronyme.html - 36k - Cached - Similar pages

[PDF] Microsoft PowerPoint - lecture9.ppt
File Format: PDF/Adobe Acrobat - View as HTML
... RPC) High level representation of **IPC** Uses the ... may block waiting for results **Synchronous** (wait) vs ... RPC Transaction processing **monitors** (transaction management ...
www.deri.at/teaching/lectures/ summer04/documents/lecture9_6up.pdf - Similar pages

New! Get the latest web results on **IPC monitor synchronous semantic** emailed to you with Google Web Alerts.

Gooooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

IPC monitor synchronous semant Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Published before July 2000
Terms used **IPC monitor block source destination**

Found **68** of **104,431**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 68 Result page: **1** 2 3 4 next

Relevance scale

1 Monitoring distributed systems
Jeffrey Joyce, Greg Lomow, Konrad Slind, Brian Unger
March 1987 **ACM Transactions on Computer Systems (TOCS)**, Volume 5 Issue 2

Full text available: pdf(2.37 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The monitoring of distributed systems involves the collection, interpretation, and display of information concerning the interactions among concurrently executing processes. This information and its display can support the debugging, testing, performance evaluation, and dynamic documentation of distributed systems. General problems associated with monitoring are outlined in this paper, and the architecture of a general purpose, extensible, distributed monitoring system is presented. Three a ...

2 The integration of virtual memory management and interprocess communication in Accent
Robert Fitzgerald, Richard F. Rashid
May 1986 **ACM Transactions on Computer Systems (TOCS)**, Volume 4 Issue 2

Full text available: pdf(2.45 MB) Additional Information: full citation, abstract, references, citings, index terms

The integration of virtual memory management and interprocess communication in the Accent network operating system kernel is examined. The design and implementation of the Accent memory management system is discussed and its performance, both on a series of message-oriented benchmarks and in normal operation, is analyzed in detail.

3 Experimental evaluation of SUNOS IPC and TCP/IP protocol implementation
Christos Papadopoulos, Gurudatta M. Parulkar
April 1993 **IEEE/ACM Transactions on Networking (TON)**, Volume 1 Issue 2

Full text available: pdf(1.75 MB) Additional Information: full citation, references, citings, index terms

4 Improving IPC by kernel design
Jochen Liedtke
December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5

Full text available: pdf(1.39 MB) Additional Information: full citation, abstract, references, citings, index terms

Inter-process communication (ipc) has to be fast and effective, otherwise programmers will not use remote procedure calls (RPC), multithreading and multitasking adequately. Thus ipc performance is vital for modern operating systems, especially μ-kernel based ones. Surprisingly, most μ-kernels exhibit poor ipc performance, typically requiring 100 μs for a short message transfer on a modern processor, running with 50 MHz clock rate.In contrast, we achieve 5 μs; a twenty ...

5 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

6 A system for interprocess communication in a resource sharing computer network

David C. Walden

April 1972 **Communications of the ACM**, Volume 15 Issue 4

Full text available: pdf(1.02 MB) Additional Information: full citation, abstract, references, citings

A system of communication between processes in a time-sharing system is described and the communication system is extended so that it may be used between processes distributed throughout a computer network. The hypothetical application of the system to an existing network is discussed.

Keywords: computer networks, interprocess communication, resource sharing, time-sharing

7 Computer Communication Networks: Approaches, Objectives, and Performance Considerations

Stephen R. Kimbleton, G. Michael Schneider

September 1975 **ACM Computing Surveys (CSUR)**, Volume 7 Issue 3

Full text available: pdf(3.99 MB) Additional Information: full citation, references, citings, index terms

8 The implementation of dynamite: an environment for migrating PVM tasks

K. A. Iskra, F. van der Linden, Z. W. Hendrikse, B. J. Overeinder, G. D. van Albada, P. M. A. Sloot

July 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 3

Full text available: pdf(1.60 MB) Additional Information: full citation, abstract, references, citings

Parallel programming on clusters of workstations is increasingly attractive, but dynamic load balancing is needed to make efficient use of the available resources. Dynamite provides dynamic load balancing for PVM applications running under Linux and Solaris. It supports migration of individual tasks between nodes in a manner transparent both to the application programmer and to the user, implemented entirely in user space. Dynamically linked executables are supported, as are tasks with open file ...

Keywords: PVM, cluster computing, message-passing, task migration

9 Flexible control of downloaded executable content
Trent Jaeger, Atul Prakash, Jochen Liedtke, Nayeem Islam
May 1999 **ACM Transactions on Information and System Security (TISSEC)**, Volume 2 Issue 2

Full text available: pdf(297.79 KB) Additional Information: full citation, abstract, references, citings, index terms, review

We present a security architecture that enables system and application a ccess control requirements to be enforced on applications composed from downloaded executable content. Downloaded executable content consists of messages downloaded from remote hosts that contain executables that run, upon receipt, on the downloading principal's machine. Unless restricted, this content can perform malicious actions, including accessing its downloading principal's private data and sending messages on th ...

Keywords: access control models, authentication, autorization machanisms, collaborative systems, role-based access control

10 The V distributed system
David Cheriton
March 1988 **Communications of the ACM**, Volume 31 Issue 3

Full text available: pdf(2.55 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The V distributed System was developed at Stanford University as part of a research project to explore issues in distributed systems. Aspects of the design suggest important directions for the design of future operating systems and communication systems.

11 Implementation trade-offs in using a restricted data flow architecture in a high performance RISC microprocessor
M. Simone, A. Essen, A. Ike, A. Krishnamoorthy, T. Maruyama, N. Patkar, M. Ramaswami, M. Shebanow, V. Thirumalaiswamy, D. Tovey
May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available: pdf(1.04 MB) Additional Information: full citation, abstract, references, citings, index terms

The implementation of a superscalar, speculative execution SPARC-V9 microprocessor incorporating Restricted Data Flow principles required many design trade-offs. Consideration was given to both performance and cost. Performance is largely a function of cycle time and instructions executed per cycle while cost is primarily a function of die area. Here we describe our Restricted Data Flow implementation and the means with which we arrived at its configuration. Future semiconductor technology advan ...

12 TCP extensions for space communications
Robert C. Durst, Gregory J. Miller, Eric J. Travis
October 1997 **Wireless Networks**, Volume 3 Issue 5

Full text available: pdf(375.24 KB) Additional Information: full citation, abstract, references, citings, index terms

The space communication environment and mobile and wireless communication environments show many similarities when observed from the perspective of a transport protocol. Both types of environments exhibit loss caused by data corruption and link outage,

in addition to congestion-related loss. The constraints imposed by the two environments are also similar—power, weight, and physical volume of equipment are scarce resources. Finally, it is not uncommon for communication channel data ra ...

13 A continuous media transport and orchestration service
Andrew Campbell, Geoff Coulson, Francisco García, David Hutchison
October 1992 **ACM SIGCOMM Computer Communication Review , Conference proceedings on Communications architectures & protocols**, Volume 22 Issue 4

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

The desire to transfer continuous media such as digital audio and video across packet switched networks imposes a number of new requirements on transport level communication services. This paper identifies a number of these requirements in the context of an experimental distributed multimedia infrastructure, and reports on research which addresses some of the associated issues. Particular attention is paid to two areas: (i) extended Quality of Service (QoS) provision; and (ii) support for t ...

14 TCP extensions for space communications
Robert C. Durst, Gregory J. Miller, Eric J. Travis
November 1996 **Proceedings of the 2nd annual international conference on Mobile computing and networking**

Full text available: pdf(1.58 MB) Additional Information: full citation, references, citings, index terms

15 The VMP network adapter board (NAB): high-performance network communication for multiprocessors
H. Kanakia, D. Cheriton
August 1988 **ACM SIGCOMM Computer Communication Review , Symposium proceedings on Communications architectures and protocols**, Volume 18 Issue 4

Full text available: pdf(1.63 MB) Additional Information: full citation, abstract, references, citings, index terms

High performance computer communication between multiprocessor nodes requires significant improvements over conventional host-to-network adapters. Current host-to-network adapter interfaces impose excessive processing, system bus and interrupt overhead on a multiprocessor host. Current network adapters are either limited in function, wasting key host resources such as the system bus and the processors, or else intelligent but too slow, because of complex transport protocols and because of a ...

16 Alternative software architectures for parallel protocol execution with synchronous IPC
C. Murray Woodside, R. Greg Franks
April 1993 **IEEE/ACM Transactions on Networking (TON)**, Volume 1 Issue 2

Full text available: pdf(963.77 KB) Additional Information: full citation, references, citings, index terms

17 Client-server computing
Alok Sinha
July 1992 **Communications of the ACM**, Volume 35 Issue 7

Full text available: pdf(7.53 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: client-server computing

18 Performance counters and state sharing annotations: a unified approach to thread locality

Boris Weissman

October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33 , 32 Issue 11 , 5

Full text available: pdf(1.76 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a combined approach for improving thread locality that uses the bardware performance monitors of modem processors and program-centric code annotations to guide thread scheduling on SMPs. The approach relies on a shared state cache model to compute expected thread footprints in the cache on-line. The accuracy of the model has been analyzed by simmations involving a set of parallel applications. We demonstrate how the cache model can be used to implement several practical loca ...

19 Instruction path coprocessors

Yuan Chou, John Paul Shen

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture**, Volume 28 Issue 2

Full text available: pdf(134.64 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents the concept of an Instruction Path Coprocessor (I-COP), which is a programmable on-chip coprocessor, with its own mini-instruction set, that operates on the core processor's instructions to transform them into an internal format that can be more efficiently executed. It is located off the critical path of the core processor to ensure that it does not negatively impact the core processor's cycle time or pipeline depth. An I-COP is highly versatile and can be used ...

20 A survey of commercial parallel processors

Edward Gehringer, Janne Abullarade, Michael H. Gulyn

September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4

Full text available: pdf(2.96 MB) Additional Information: full citation, abstract, citings, index terms

This paper compares eight commercial parallel processors along several dimensions. The processors include four shared-bus multiprocessors (the Encore Multimax, the Sequent Balance system, the Alliant FX series, and the ELXSI System 6400) and four network multiprocessors (the BBN Butterfly, the NCUBE, the Intel iPSC/2, and the FPS T Series). The paper contrasts the computers from the standpoint of interconnection structures, memory configurations, and interprocessor communication. Also, the share ...

Results 1 - 20 of 68 Result page: **1** 2 3 4 next